

## REMARKS

### Rejection of claims 1-7 and 16-20 under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of JDK

The Examiner rejected claims 1-7 and 16-20 under 35 U.S.C. §103(a) as being unpatentable over De Borst in view of JDK. Each of the independent claims in this group, namely claims 1, 7 and 16, have been amended to recite that the datastream receive mechanism is defined in the datastream class, and causes the instance of the datastream class to populate itself with information contained in the datastream by invoking at least one object method on the instance. Both De Borst and JDK teach datastream objects that include methods (such as get() and put() for De Borst, and writeObject() and readObject() for JDK) that allow writing to or reading from a datastream. However, nowhere does De Borst nor JDK teach or suggest intelligence within the datastream class that allows the instance to populate itself from information in the datastream, as currently recited in claims 1, 7 and 16. For this reason, claims 1, 7 and 16 as amended are clearly allowable over the combination of De Borst and JDK.

Claims 2-6 depend on claim 1, which is allowable for the reasons given above. Claims 17-20 depend on claim 16, which is allowable for the reasons given above. As a result, claims 2-6 and 17-20 are allowable as depending on allowable independent claims.

Rejection of claims 8-15 under 35 U.S.C. §103(a) as being unpatentable over the combination of De Borst, JDK and Sosic


The Examiner rejected claims 8-15 under 35 U.S.C. §103(a) as being unpatentable over the combination of De Borst, JDK and Sosic. The independent claims in this group, namely claims 8, 11 and 15, have been amended herein to recite that the instance of the datastream class populates itself with information contained in the active datastream by invoking at least one method on the new instance. None of De Borst, JDK or Sosic teaches or suggests an instance of a datastream class that can populate itself with information contained in the datastream by invoking at least one method on the datastream instance. For this reason, each of independent claims 8, 11 and 15 are allowable over the combination of De Borst, JDK and Sosic.

Claims 9-10 depend on claim 8, which is allowable for the reasons given above. Claims 12-14 depend on claim 11, which is allowable for the reasons given above. As a result, each of claims 9-10 and 12-14 are allowable as depending on allowable independent claims.

Conclusion

In summary, neither De Borst, JDK, Sasic, nor their combination teach, support, or suggest the unique combination of features in applicants' claims presently on file. Therefore, applicants respectfully assert that all of applicants' claims are allowable. Such allowance at an early date is respectfully requested. The Examiner is invited to telephone the undersigned if this would in any way advance the prosecution of this case.

Respectfully submitted,

By   
Derek P. Martin  
Reg. No. 36,595

**MARTIN & ASSOCIATES, L.L.C.**  
P.O. Box 548  
Carthage, MO 64836-0548  
(417) 358-4700